

|              |              |       |
|--------------|--------------|-------|
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAAAA |
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAAAA |
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAAAA |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSSSSSSSSS   | DDD          | AAA   |
| SSSSSSSSSS   | DDD          | AAA   |
| SSSSSSSSSS   | DDD          | AAA   |
| SSS          | DDD          | AAAAA |
| SSS          | DDD          | AAAAA |
| SSS          | DDD          | AAAAA |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSS          | DDD          | AAA   |
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAA   |
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAA   |
| SSSSSSSSSSSS | DDDDDDDDDDDD | AAA   |



|          |          |        |            |            |            |    |          |          |
|----------|----------|--------|------------|------------|------------|----|----------|----------|
| SSSSSSSS | DDDDDDDD | AAAAAA | DDDDDDDD   | EEEEEEEEEE | FFFFFFFFFF |    |          |          |
| SSSSSSSS | DDDDDDDD | AAAAAA | DDDDDDDD   | EEEEEEEEEE | FFFFFFFFFF |    |          |          |
| SS       | DD       | DD     | AA         | AA         | DD         | DD | EE       | FF       |
| SS       | DD       | DD     | AA         | AA         | DD         | DD | EE       | FF       |
| SS       | DD       | DD     | AA         | AA         | DD         | DD | EE       | FF       |
| SS       | DD       | DD     | AA         | AA         | DD         | DD | EE       | FF       |
| SSSSSS   | DD       | DD     | AA         | AA         | DD         | DD | EEEEEEEE | FFFFFFFF |
| SSSSSS   | DD       | DD     | AA         | AA         | DD         | DD | EEEEEEEE | FFFFFFFF |
| SS       | DD       | DD     | AAAAAAAAAA | DD         | DD         | DD | EE       | FF       |
| SS       | DD       | DD     | AAAAAAAAAA | DD         | DD         | DD | EE       | FF       |
| SS       | DD       | DD     | AA         | AA         | DD         | DD | EE       | FF       |
| SS       | DD       | DD     | AA         | AA         | DD         | DD | EE       | FF       |
| SSSSSSSS | DDDDDDDD | AA     | AA         | DDDDDDDD   | EEEEEEEEEE | FF | ....     |          |
| SSSSSSSS | DDDDDDDD | AA     | AA         | DDDDDDDD   | EEEEEEEEEE | FF | ....     |          |
|          |          |        |            |            |            |    | ....     |          |
|          |          |        |            |            |            |    | ....     |          |

|          |          |            |    |
|----------|----------|------------|----|
| SSSSSSSS | DDDDDDDD | LL         |    |
| SSSSSSSS | DDDDDDDD | LL         |    |
| SS       | DD       | DD         | LL |
| SS       | DD       | DD         | LL |
| SS       | DD       | DD         | LL |
| SS       | DD       | DD         | LL |
| SSSSSS   | DD       | DD         | LL |
| SSSSSS   | DD       | DD         | LL |
| SS       | DD       | DD         | LL |
| SS       | DD       | DD         | LL |
| SS       | DD       | DD         | LL |
| SS       | DD       | DD         | LL |
| SSSSSSSS | DDDDDDDD | LLLLLLLLLL |    |
| SSSSSSSS | DDDDDDDD | LLLLLLLLLL |    |



{ Title SDADEF - System Dump Analyzer Internal Definitions

{ Version: 'V04-000'

{\*\*\*\*\*  
{\*  
{\* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
{\* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
{\* ALL RIGHTS RESERVED.  
{\*  
{\* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
{\* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
{\* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
{\* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
{\* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
{\* TRANSFERRED.  
{\*  
{\* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
{\* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
{\* CORPORATION.  
{\*  
{\* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
{\* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
{\*  
{\*\*\*\*\*

{++

{ FACILITY: SDA Macro Library

{ ABSTRACT:

{ This file contains the SDL source for all System Dump Analyzer  
{ structure blocks.

{ ENVIRONMENT:

{ Used by the System Dump Analyzer which is a user mode image capable  
{ of switching to kernel or any other mode.

{--

{ AUTHOR: Ralph O. Weber CREATION DATE: 23-OCT-1983

{ MODIFIED BY:

{ V03-008 PRB0303 Paul Beck 28-Dec-1983 17:53  
{ Add OPT symbol for /NOSKIP qualifier on EXAM/INST  
{  
{ V03-007 ROW0237 Ralph O. Weber 23-OCT-1983  
{ Combine SDADEF.MDL and SCRDEF.MDL to produce this SDL file.  
{ Also add structure definitions for the column output routine,  
{ \$COLMDEF.  
{



V03-006 WMC0001 Wayne Cardoza 11-Jul-1983  
OPT symbols for P0 and P1 process page tables.

V03-005 CWH1002 CW Hobbs 13-Mar-1983  
Add OPT symbol for show summary /image

V03-004 RAS0123 Ron Schaefer 8-Feb-1983  
Add OPT symbol for show/rms=rjb.

V03-003 JLV0225 Jake VanNoy 21-JAN-1983  
Add OPT symbols for ex/cond, ex/time, ex/psl and validate  
queue.

V03-002 LMP0031 L. Mark Pilant, 17-Jun-1982 13:07  
Add support for a new command: SHOW PROCESS/CHANNELS.

V03-001 MSH0001 Maryann Hinden 10-Jun-1982  
Add LCK option for SHOW PROC.

```
{++
{
{ $COLMDEF -- Structure of an entry in a column definition list
{ This is the structure on a single entry in the list which describes
{ a column in a multi-column table. The list is constructed by the
{ COLUMN_LIST macro and processed by the PRINT_COLUMNS routine.
{
{--

module $COLMDEF;

aggregate COLMDEF structure prefix COLM$:

  STRING longword unsigned;          /* Address of text for description column
  SOURCE longword;                   /* Description of data source
  ACTION_VALUE union longword unsigned; /* Value passed to action routine
    SRC_FAO byte unsigned;           /* Data type FAO selector for SOURCE
    constant (
      FAO_AC, FAO_AS,                { Strings
      FAO_OB, FAO_XB, FAO_ZB, FAO_UB, FAO_SB, { Bytes
      FAO_OW, FAO_XW, FAO_ZW, FAO_UW, FAO_SW, { Words
      FAO_OL, FAO_XL, FAO_ZL, FAO_UL, FAO_SL { Longwords
    ) equals 0 increment 1;
    { now the display if not equal values
    constant (
      FAO_OB_NEQ, FAO_XB_NEQ, FAO_ZB_NEQ, FAO_UB_NEQ, FAO_SB_NEQ,
      FAO_OW_NEQ, FAO_XW_NEQ, FAO_ZW_NEQ, FAO_UW_NEQ, FAO_SW_NEQ,
      FAO_OL_NEQ, FAO_XL_NEQ, FAO_ZL_NEQ, FAO_UL_NEQ, FAO_SL_NEQ
    ) equals COLM$K_FAO_OB+80 increment 1;
    { finally, the "special case" codes
    constant (
      FAO_Q2                          { a doubly-linked queue header
    ) equals COLM$K_FAO_SL+1 increment 1;
  end ACTION_VALUE;
  DESC_SIZE byte unsigned;           /* Size of descriptor column
  VAL_SIZE byte unsigned;            /* Size of value column
  SEP_SIZE byte unsigned;            /* Size of separator column
  RESERVED byte fill;
  constant "LENGTH" equals .;        /* Length of one entry

end COLMDEF;

end_module $COLMDEF;
```



```
{++
{ $NODEDEF -- Symbol Table Tree Node Definitions
{--
module $NODEDEF;
aggregate NODEDEF structure prefix NODE$;
    LEFT longword unsigned;      /* Left subtree pointer
    RIGHT longword unsigned;     /* Right subtree pointer
    BAL word unsigned;           /* Balance at this node (-1,0,1)
    PTR longword unsigned;       /* Pointer to symbol table entry (SYM$)
    constant 'LENGTH' equals . tag C; /* Length of node
end NODEDEF;
end_module $NODEDEF;
```

```

{++
{ $OBJDEF -- OBJECT MODULE DEFINITIONS
{ THIS MODULE CONTAINS DEFINITIONS FOR THE FIELDS IN AN OBJECT
{ MODULE RECORD.
{--

module $OBJDEF;

aggregate OBJDEF structure prefix OBJ$;

    TYPE_OVERLAY union fill;
        TYPE byte unsigned;
            constant (
                HDR,
                GSD,
                TIR,
                EOM,
                DBG,
                TBT,
                LNK
            ) equals 0 increment 1 tag C;
        GSD_TYPE byte unsigned;
            constant (
                GSD_PSC,
                GSD_SYM,
                GSD_EPM,
                GSD_PRO
            ) equals 0 increment 1 tag C;
    end TYPE_OVERLAY;

    RECORDS_OVERLAY union fill;

{
{ FORMAT FOR PSECT DEFINITION
{
    PSECT_RECORD structure fill;
        PSC_ALI byte unsigned;
        PSC_FLAG union word unsigned;
            PSC_PIC bitfield mask;
            PSC_LIB bitfield mask;
            PSC_OVL bitfield mask;
            PSC_REL bitfield mask;
            PSC_GBL bitfield mask;
            PSC_SHR bitfield mask;
            PSC_EXE bitfield mask;
            PSC_RD bitfield mask;
            PSC_WRT bitfield mask;
        end PSC_FLAG;
        PSC_LEN longword unsigned;
        PSC_NAME character;
    end PSECT_RECORD;

        /* Type of record:
        /* Header
        /* Global symbol
        /* Text and information
        /* End of module
        /* Debug
        /* Traceback
        /* Link options

        /* Type of GSD record:
        /* PSECT definition
        /* Global symbol definition
        /* Entry point definition
        /* Procedure definition

        /* PSECT alignment (2**n)
        /* PSECT flags:
        /* Position independent
        /* Defined from sharable image
        /* Overlaid psect
        /* Requires relocation
        /* PSECT is global
        /* Potentially sharable
        /* Executable
        /* Can be read
        /* Can be written

        /* Length of PSECT
        /* PSECT name (counted string)

```



```
{
{
{
FORMAT FOR GLOBAL SYMBOL DEFINITIONS
{
```

```
GLOBAL_SYM_RECORD structure fill;
  SYM_DTYPE byte unsigned;          /* Data type (see Arch. Handbook, Ap. C)
  SYM_FLAGS union word unsigned;    /* Symbol flags:
    SYM_WEAK bitfield mask;        /* Weak resolution (0 = strong)
    SYM_DEF bitfield mask;         /* Definition (0 = reference)
    SYM_UNI bitfield mask;         /* Universal definition (0 = local)
    SYM_REL bitfield mask;         /* Relative symbol (0 = absolute)
  end SYM_FLAGS;
  SYM_PSIND byte unsigned;          /* PSECT index (only if def)
  SYM_VALUE longword unsigned;      /* Value of symbol (only if def)
  SYM_NAME character;              /* Symbol name (counted string)
end GLOBAL_SYM_RECORD;
```

```
{
{
{
FORMAT FOR ENTRY POINT RECORDS
{
```

```
ENTRY_POINT_RECORD structure fill;
  EPM_DTYPE byte unsigned;          /* Data type (see Arch. Handbook, Ap. C)
  EPM_FLAGS word unsigned;          /* Flags (same as SYM_FLAGS)
  EPM_PSIND byte unsigned;          /* PSECT index (only if def)
  EPM_VALUE longword unsigned;      /* Value of symbol (only if def)
  EPM_MASK word unsigned;           /* Entry mask
  EPM_NAME character;              /* Symbol name (counted string)
end ENTRY_POINT_RECORD;
```

```
{
{
{
FORMAT FOR END OF MODULE RECORD
{
```

```
EOM_RECORD structure fill;
  EOM_SEV byte unsigned;            /* Error severity for module
  EOM_PSIND byte unsigned;          /* PSECT index of transfer address
  EOM_TRANS longword unsigned;      /* Transfer address
end EOM_RECORD;
```

```
end RECORDS_OVERLAY;
```

```
end OBJDEF;
```

```
end_module $OBJDEF;
```



```
{++
{
$OPTDEF -- Options Selected From Commands
{--

module $OPTDEF;

aggregate OPTDEF union prefix OPT$;

  SHOW PROCESS structure fill;
    WSL bitfield mask;
    PPT bitfield mask;
    PST bitfield mask;
    REGS bitfield mask;
    PCB bitfield mask;
    PHD bitfield mask;
    SYSPROC bitfield mask;
    RMS bitfield mask;
    RMSD bitfield mask;
    LCK bitfield mask;
    CHAN bitfield mask;
    PO_PPT bitfield mask;
    P1_PPT bitfield mask;
    PPT_LEN bitfield mask;
    PPT_RNG bitfield mask;
  end SHOW_PROCESS;

  SHOW POOL structure fill;
    FREE bitfield mask;
    IRP bitfield mask;
    NONPAGED bitfield mask;
    PAGED bitfield mask;
    'LENGTH' bitfield mask;
    LRP bitfield mask;
    SUMMARY bitfield mask;
    HEADER bitfield mask;
    TYPE bitfield mask;
    SRP bitfield mask;
  end SHOW_POOL;

  EXAMINE structure fill;
    P0 bitfield mask;
    P1 bitfield mask;
    SYSTEM bitfield mask;
    RANGE bitfield mask;
    LENGTH bitfield mask;
    INST bitfield mask;
    PSL bitfield mask;
    TIME bitfield mask;
    COND bitfield mask;
    NOSKIP bitfield mask;
  end EXAMINE;

  VALIDATE structure fill;
    SELF bitfield mask;
```

```
{ --- SHOW PROCESS options
/* WORKING SET LIST
/* PROCESS PAGE TABLES
/* PROCESS SECTION TABLE
/* REGISTERS
/* PROCESS CONTROL BLOCK
/* PROCESS HEADER
/* /SYSTEM ('SYSTEM PROCESS')
/* RMS STRUCTURES
/* RMS STRUCTURES WITH DISPLAY
/* LOCK DATA STRUCTURES
/* PROCESS CHANNELS
/* P0 PROCESS PAGE TABLE
/* P1 PROCESS PAGE TABLE
/* PAGE TABLE LENGTH SPECIFIED
/* PAGE TABLE RANGE SPECIFIED

{ --- SHOW POOL options
/* SHOW FREE HOLES
/* IRP LOOKASIDE LIST
/* NON-PAGED DYNAMIC POOL
/* PAGED DYNAMIC POOL
/* LENGTH SPECIFIED
/* LRP LOOKASIDE LIST
/* SUMMARY STATISTICS
/* ONLY WANT HEADER
/* TYPE SPECIFIED
/* SRP LOOKASIDE LIST

{ --- EXAMINE options
/* P0 SPACE
/* P1 SPACE
/* SYSTEM SPACE
/* RANGE SPECIFIED (start:end)
/* LENGTH SPECIFIED (start:length)
/* EXAMINE/INSTRUCTION
/* EXAMINE/PSL
/* EXAMINE/TIME
/* EXAMINE/CONDITION VALUE
/* EXAMINE/INSTRUCTION/NOSKIP

{ --- VALIDATE options
/* SELF RELATIVE QUEUE
```

end VALIDATE;

SHOW\_STACK structure fill;

ISP bitfield mask;  
KSP bitfield mask;  
ESP bitfield mask;  
SSP bitfield mask;  
USP bitfield mask;

end SHOW\_STACK;

SHOW\_PFN structure fill;

FREE bitfield mask;  
MODIFIED bitfield mask;  
BAD bitfield mask;  
WHOLEPFN bitfield mask;  
SINGLEPFN bitfield mask;

end SHOW\_PFN;

SHOW\_PAGE structure fill;

GLOBAL bitfield mask;

end SHOW\_PAGE;

SHOW\_SUMMARY structure fill;

IMAGE bitfield mask;

end SHOW\_SUMMARY;

SET\_RMS structure fill;

NO bitfield mask;  
IFB bitfield mask;  
IRB bitfield mask;  
IDX bitfield mask;  
BDB bitfield mask;  
BDBSUM bitfield mask;  
ASB bitfield mask;  
CCB bitfield mask;  
WCB bitfield mask;  
FCB bitfield mask;  
FAB bitfield mask;  
RAB bitfield mask;  
NAM bitfield mask;  
XAB bitfield mask;  
RLB bitfield mask;  
BLB bitfield mask;  
BLBSUM bitfield mask;  
GBD bitfield mask;  
GBH bitfield mask;  
TRC bitfield mask;  
FWA bitfield mask;  
GBDSUM bitfield mask;  
RJB bitfield mask;

constant RMSALL equals -2 tag M;  
constant ALL equals 4095 tag M;

end SET\_RMS;

{ --- SHOW STACK options

/\* INTERRUPT STACK  
/\* KERNEL STACK  
/\* EXECUTIVE STACK  
/\* SUPERVISOR STACK  
/\* USER STACK

{ --- SHOW PFN DATA options

/\* FREE PAGE LIST  
/\* PAGE LIST  
/\* BAD PAGE LIST  
/\* PFN DATA BASE  
/\* PFN

{ --- SHOW PAGE TABLE options

{ SYSTEM PAGE TABLE (USE EXISTING ONE)  
/\* GLOBAL PAGE TABLE

{ --- SHOW SUMMARY options

/\* IMAGE FILE NAME

{ --- SET RMS options

/\* NEGATE THIS OPTION (MUST BE = 1)

/\* IFAB  
/\* IRAB  
/\* IDX  
/\* BDB  
/\* BDB SUMMARY  
/\* ASB  
/\* CCB  
/\* WCB  
/\* FCB  
/\* FAB  
/\* RAB  
/\* NAM  
/\* XAB  
/\* RLB  
/\* BLB  
/\* BLB SUMMARY  
/\* GBD  
/\* GBH  
/\* TRC  
/\* FWA  
/\* GBD SUMMARY  
/\* RJB

/\* UPPER 31 BITS ON  
/\* 12 BITS ON



SDADEF.SDL;1

16-SEP-1984 16:44:48.85 <sup>K 11</sup> Page 9

end OPTDEF;

end\_MODULE \$OPTDEF;

\*\*

```
{++
{
{ $SCRDEF -- Definitions for the SDA to screen package interface
{--
module $SCRDEF;
aggregate SCRDEF structure prefix SCR$;
    FLAGS union longword unsigned;
        SCREEN bitfield mask;
    end FLAGS;
    WIDTH word unsigned;
    PAGESIZE word unsigned;
    DEVTYPE byte unsigned;
    RESERVED byte dimension 11 fill;
    constant "LENGTH" equals . tag C;
end SCRDEF;
end_module $SCRDEF;
```



```
{++
{
{ $SYMDEF -- SDA Symbol Table Definitions
{ Definitions for a dump analyzer symbol table entry.
{--
module $SYMDEF;
aggregate SYMDEF structure prefix SYM$;
    VALUE longword unsigned;          /* Value of symbol
    SYMBOL character length 32;        /* Symbol name (counted string)
    constant 'LENGTH' equals . tag C; /* Length of an entry
end SYMDEF;
end_module $SYMDEF;
```



0350

AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY